Mathematical Programming
https://doi.org/10.1007/s10107-020-01609-8

FULL LENGTH PAPER

Series B

Generalized adaptive partition-based method for two-stage stochastic linear programs with fixed recourse

Cristian Ramirez-Pico[1] · **Eduardo Moreno**[1]

Received: 3 March 2020 / Accepted: 11 December 2020

Abstract
We present a method to solve two-stage stochastic linear programming problems with fixed recourse when the uncertainty space can have either discrete or continuous distributions. Given a partition of the uncertainty space, the method is addressed to solve a discrete problem with one scenario for each element of the partition (subregions of the uncertainty space). Fixing first-stage variables, we formulate a second-stage subproblem for each element, and exploiting information from the dual of these problems, we provide conditions that the partition must satisfy to obtain an optimal solution. These conditions provide guidance on how to refine the partition, iteratively approaching an optimal solution. The results from computational experiments show how the method automatically refines the partition of the uncertainty space in the regions of interest for the problem. Our algorithm is a generalization of the adaptive partition-based method presented by Song and Luedtke for discrete distributions, extending its applicability to more general cases.

Keywords Two-stage stochastic programming · Adaptive partition-based approach · Continuous distribution · Scenario aggregation

Mathematics Subject Classification 90C15 · 90-08 · 65K05

Supported by ANID/CONICYT-Fondecyt Regular 1161064 and 1200809.

✉ Eduardo Moreno
eduardo.moreno@uai.cl

Cristian Ramirez-Pico
cristian.ramirez@edu.uai.cl

[1] Faculty of Engineering and Sciences, Universidad Adolfo Ibáñez, Santiago, Chile

Published online: 23 February 2021

1 Introduction

We study the following two-stage stochastic program (TSSP) with fixed recourse

$$\min \left\{ c^\top x + \mathbb{E}\left[\mathcal{Q}(x,\xi)\right] \mid x \in \mathcal{X} \right\} \tag{1}$$

where $\mathcal{X} \subseteq \mathbb{R}^n$ is a set assumed to be nonempty and closed, ξ is a random vector in the probability space $(\Omega, \mathcal{A}, \mathbb{P})$ containing the random elements $\left\{h^\xi, T^\xi\right\}$, and second-stage subproblem

$$\mathcal{Q}(x,\xi) := \min \left\{ q^\top y \mid Wy = h^\xi - T^\xi x,\ y \geq 0 \right\}, \tag{2}$$

where fixed recourse matrix $W \in \mathbb{R}^{m\times n}$, deterministic costs $q \in \mathbb{R}^n$, random technology matrix $T^\xi \in \mathbb{R}^{m\times n}$ and random right-hand side vector $h^\xi \in \mathbb{R}^m$. Furthermore, we assume that there exists some $\bar{x} \in \mathcal{X}$ such that $\mathcal{Q}(\bar{x},\xi)$ is feasible and bounded in the whole uncertainty space Ω. Without loss of generality, we assume $\mathcal{Q}(x,\xi) = \infty$ if the problem is infeasible. Note that the support of the uncertainty space Ω can be either continuous or discrete.

In this paper, we propose a method to solve TSSPs by iteratively and automatically aggregating the uncertainty space into a small number of scenarios and disaggregating them based on the information of dual subproblem variables. This approach yields an exact reformulation of the original stochastic programming problem (1) but with significantly fewer variables and constraints. For the case of discrete distributions, this idea has been called the *adaptive partition-based method* (APM) by Song and Luedtke [34], and it is based on the results of Espinoza and Moreno [11] and Bienstock and Zuckerberg [5]. We present an alternative proof that allows us to extend APM to a more general setting, particularly to address TSSPs with continuous distributions for Ω.

Let $P \subseteq \Omega$, and let $T^P = \mathbb{E}\left[T^\xi | P\right]$ and $h^P = \mathbb{E}\left[h^\xi | P\right]$ be the conditional expectations of the components of ξ given P. We denote the *aggregated subproblem* as

$$\mathcal{Q}(x, \mathbb{E}\left[\xi | P\right]) = \min \left\{ q^\top y \mid Wy = h^P - T^P x,\ y \geq 0 \right\} \tag{3}$$

The contribution of this paper is to provide conditions for a partition $\mathcal{P}$ of Ω such that the solution of Problem (1) is equivalent to solving

$$\min_{x\in\mathcal{X}} \left\{ c^\top x + \sum_{P\in\mathcal{P}} \mathcal{Q}\left(x, \mathbb{E}\left[\xi | P\right]\right) \cdot \mathbb{P}(P) \right\}. \tag{4}$$

Note that this problem is equivalent to a TSSP with a discrete distribution of $|\mathcal{P}|$ scenarios for the uncertainty space. Moreover, this approach enables us to generate algorithms to obtain exact optimal solutions for general TSSPs.

The remainder of this paper is organized as follows. Section 2 reviews the literature concerning the APM for discrete TSSPs and other approaches to solve this problem. Section 3 develops the generalized adaptive partition-based method, with the main

mathematical results to validate this approach. Section 4 discusses the details of the algorithms that are implemented for two well-known stochastic programming problems in Sect. 5. Finally, concluding remarks are presented in Section 6.

2 Literature review

In past decades, researchers have developed solution strategies for multiple stochastic optimization problems. Although most studies start from the deterministic equivalent formulation to obtain alternative models that are more tractable in algorithmic terms, one of the most studied and utilized types of problems is two-stage stochastic programming problems. In their seminal paper, Kleywegt et al. [15] showed that any TSSP formulation can be approximated by solving Problem (1) for a discrete set of samples of ξ from the original uncertainty space Ω: they called this result the sample average approximation (SAA) method. A key fact from their paper is that good approximations require a large number of scenarios to guarantee an ϵ-optimal solution. Since then, most research on this problem has been focused on solving large-scale instances of discrete TSSPs with many scenarios.

A common and widely studied approach is to decompose TSSPs via the block structure of the scenario formulation. The most classic approach is called Benders decomposition (or the L-Shaped method as its stochastic variant [36]).

Most improvements of this approach focused on reducing the algorithm instability, such as the case of regularized decomposition [30], level decomposition [17,40] and inexact bundle methods [22,39]. Recent developments with respect to Benders decomposition are proposed in [25–27], which primarily explore how to accelerate and parallelize the technique, and [1,31], which consider how to address integer problems.

Other decomposition methodologies include stochastic decomposition [12], progressive hedging [29,38] and stochastic dual dynamic programming [24] for the case of multistage stochastic programming problems.

A different decomposition approach was developed based on the general decomposition method proposed by Bienstock and Zuckerberg [5,21]. Espinoza and Moreno [11] introduced an algorithm based on this decomposition method to minimize risk measures in linear programs. This idea was later extended by Song and Luedtke [34] to general TSSPs with discrete distributions, where the term *adaptive partition-based method* was coined. These studies have been extended recently by combination with Benders decomposition [23] and level decomposition [23,35], and new extensions have been made to multistage stochastic programming problems [33].

As mentioned previously, most recent developments are oriented to the discrete case, relying on approximation by samples of continuous probability distributions for uncertain parameters. Exact methods for TSSPs with nondiscrete distributions are scarce in the recent literature, and they focus mostly on particular problems and distributions that can be reformulated in a more tractable manner. For example, [3] introduced equivalent linear and nonlinear formulations for TSSPs with simple recourse according to the probability distributions of random parameters, and [8] posed a methodology that benefits from the reduced cost of duality and sensitivity analysis to

fix the correct values of some variables in the stochastic program, thereby reducing the size of the original problem. Several bounds for TSSPs also have appeared in the classic literature of stochastic programming. Birge and Wets [6] studied different approximation schemes for stochastic programs with recourse, particularly providing the lower bound of the problem utilized in our method. Edirisinghe and Ziemba [9] derived an upper bound for the problem based on the Edmundson–Madansky inequality [10,20] that assumes boundedness and stochastic independence between the components of the random variable. This idea was subsequently extended for linear stochastic programming problems by Kall and Mayer [14] by partitioning the uncertainty space to approximate the uncertainty distribution by a finite number of aggregated scenarios. Their method, named DAPPROX, disaggregates this partition iteratively, theoretically converging to the true optimal solution. Our proposed method uses a similar idea of partitioning the uncertainty space and disaggregating them iteratively, but we define the elements of the partition in a completely different way. In fact, our method does not require an upper bound, and the partitioning is focused on improving the lower bound instead of the upper bound.

3 Generalized adaptive partition-based method

We propose a methodology that benefits from a structure shared by aggregated and atomized subproblems, which later allows us to derive conditions such that the scenarios (either finite or an infinite number of them) belonging to a certain subset of Ω, yield the same expected value of optimal solutions, as if we solve the aggregated Problem (3).

3.1 Relations between atomized and aggregated subproblems

As a first step, we define the relation between subproblems (2) and aggregated subproblems (3). It is known that the objective value of the aggregated subproblem provides a lower bound on the conditional expectation of the atomized subproblems.

Proposition 1 ([6,14]) *Let $\bar{x} \in \mathcal{X}$ such that $\mathcal{Q}(\bar{x}, \xi)$ is convex on ξ, and let $P \subseteq \Omega$. Then*

$$\mathcal{Q}\big(\bar{x}, \mathbb{E}\,[\xi \,|\, P]\big) \leq \mathbb{E}\big[\mathcal{Q}(\bar{x}, \xi) | P\big]$$

A proof of this result comes from the convexity of $\mathcal{Q}(\bar{x}, \xi)$ on ξ and Jensen's inequality. Nevertheless, a deeper connection between these two problems can be proved. Lemma 1 shows how a feasible solution of (3) can be constructed using information from an optimal solution of (2).

Lemma 1 *Let $\bar{x} \in \mathcal{X}$ and $P \subseteq \Omega$, and let $\hat{y}^{\xi}$ be an optimal solution of $\mathcal{Q}(\bar{x}, \xi)$ for $\xi \in P$. Then, $\hat{y}^{P} := \mathbb{E}\big[\hat{y}^{\xi} | P\big]$ is a feasible solution for $\mathcal{Q}(\bar{x}, \mathbb{E}\,[\xi \,|\, P])$.*

Proof Since $W\hat{y}^{\xi} = h^{\xi} - T^{\xi}\bar{x}$ for every $\xi \in P$,

$$\int_{\Omega} W\hat{y}^{\xi}\, d\mathbb{P}(\xi|P) = \int_{\Omega} \left[h^{\xi} - T^{\xi}\bar{x}\right] d\mathbb{P}(\xi|P)$$
$$W\int_{\Omega} \hat{y}^{\xi}\, d\mathbb{P}(\xi|P) = \int_{\Omega} h^{\xi}\, d\mathbb{P}(\xi|P) - \left(\int_{\Omega} T^{\xi}\, d\mathbb{P}(\xi|P)\right)\bar{x}$$
$$W\mathbb{E}\left[\hat{y}^{\xi}|P\right] = \mathbb{E}\left[h^{\xi}|P\right] - \mathbb{E}\left[T^{\xi}|P\right]\bar{x}.$$

Hence, $\hat{y}^{P}$ is a feasible solution for $\mathcal{Q}(\bar{x}, \mathbb{E}[\xi|P])$. □

Since these second-stage subproblems are bounded and consider only continuous variables, we can introduce a dual formulation for each subproblem (2) and (3), respectively,

$$\mathcal{Q}^{D}(x,\xi) := \max\left\{(h^{\xi} - T^{\xi}x)^{\top}\lambda^{\xi} \mid W^{\top}\lambda^{\xi} \le q\right\} \tag{5}$$

and

$$\mathcal{Q}^{D}(x, P) := \max\left\{(h^{P} - T^{P}x)^{\top}\lambda^{P} \mid W^{\top}\lambda^{P} \le q\right\}. \tag{6}$$

Indices ξ and P on dual variable λ distinguish between atomized and aggregated subproblems.

Similar to the primal case, we can construct a feasible solution for problem (6) based on an optimal solution of (5).

Lemma 2 *Let $\bar{x} \in \mathcal{X}$ and $P \subseteq \Omega$, and let $\hat{\lambda}^{\xi}$ be an optimal solution of problem $\mathcal{Q}^{D}(\bar{x},\xi)$ for $\xi \in P$. Then, $\hat{\lambda}^{P} := \mathbb{E}\left[\hat{\lambda}^{\xi}|P\right]$ is a feasible solution for $\mathcal{Q}^{D}(\bar{x}, P)$*

Proof Since $W^{\top}\hat{\lambda^{\xi}} \le q$ for all $\xi \in P$,

$$W^{\top}\hat{\lambda}^{P} = W^{\top}\int_{\Omega}\hat{\lambda^{\xi}}d\mathbb{P}(\xi|P) = \int_{\Omega} W^{\top}\hat{\lambda^{\xi}}d\mathbb{P}(\xi|P) \le \int_{\Omega} q\cdot d\mathbb{P}(\xi|P) = q.$$

Hence, $\hat{\lambda^{P}}$ is a feasible solution for problem $\mathcal{Q}^{D}(\bar{x}, P)$ whenever set P has positive measure. □

3.2 Construction of an optimal partition

The previous framework provides the of tools necessary to identify the conditions on P to obtain an equality between $\mathbb{E}[\mathcal{Q}(\bar{x},\xi)|P]$ and its lower bound $\mathcal{Q}(\bar{x}, \mathbb{E}[\xi|P])$.

Proposition 2 *Let $\bar{x} \in \mathcal{X}$ and $P \subseteq \Omega$, such that $\mathcal{Q}(\bar{x},\xi)$ is feasible for all $\xi \in P$, and let $\hat{\lambda}^{\xi}$ be its dual optimal solutions. If $\hat{\lambda}^{\xi}$ for $\xi \in P$ satisfies*

$$\left(\mathbb{E}\left[h^{\xi}|P\right]\right)^{\top}\left(\mathbb{E}\left[\hat{\lambda}^{\xi}|P\right]\right) = \mathbb{E}\left[h^{\xi\,\top}\hat{\lambda}^{\xi}\middle|P\right] \tag{7a}$$
$$\bar{x}^{\top}\left(\mathbb{E}\left[T^{\xi}|P\right]^{\top}\mathbb{E}\left[\hat{\lambda}^{\xi}|P\right]\right) = \bar{x}^{\top}\mathbb{E}\left[T^{\xi\,\top}\hat{\lambda}^{\xi}\middle|P\right] \tag{7b}$$

then

$$\mathcal{Q}\big(\bar{x}, \mathbb{E}\left[\xi|P\right]\big) = \mathbb{E}\big[\mathcal{Q}(\bar{x}, \xi)|P\big]$$

Proof From Proposition 1 we know that $\mathcal{Q}(\bar{x}, \mathbb{E}\left[\xi|P\right]) \leq \mathbb{E}\left[\mathcal{Q}(\bar{x}, \xi)|P\right]$. The other inequality comes from Lemma 2 and applying the conditions of the proposition.

In fact, according to Lemma 2, we know that $\hat{\lambda}^P := \mathbb{E}[\hat{\lambda}^\xi|P]$ is a feasible solution of $\mathcal{Q}^D(\bar{x}, P)$; thus,

$$\begin{aligned}\mathcal{Q}^D\big(\bar{x}, P\big) &= \mathcal{Q}(\bar{x}, \mathbb{E}\left[\xi|P\right]) \\ &\geq \Big(\mathbb{E}\big[h^\xi|P\big] - \mathbb{E}\big[T^\xi|P\big]\bar{x}\Big)^\top \Big(\mathbb{E}\big[\hat{\lambda}^\xi|P\big]\Big) \\ &= \Big(\mathbb{E}\big[h^\xi|P\big]\Big)^\top \Big(\mathbb{E}\big[\hat{\lambda}^\xi|P\big]\Big) - \bar{x}^\top \Big(\mathbb{E}\big[T^\xi|P\big]\Big)^\top \Big(\mathbb{E}\big[\hat{\lambda}^\xi|P\big]\Big)\end{aligned}$$

Since $\hat{\lambda}^P$ satisfies conditions (7), by means of the linearity of the expectation, we obtain

$$\begin{aligned}\mathcal{Q}(\bar{x}, \mathbb{E}\left[\xi|P\right]) &\geq \Big(\mathbb{E}\Big[\big(h^\xi\big)^\top \hat{\lambda}^\xi \Big| P\Big]\Big) - \bar{x}^\top \Big(\mathbb{E}\Big[\big(T^\xi\big)^\top \hat{\lambda}^\xi \Big| P\Big]\Big) \\ &= \mathbb{E}\big[\mathcal{Q}(\bar{x}, \xi)|P\big]\end{aligned}$$

□

Finally, we can establish our main results as presented in Theorem 1 and Corollary 1 as follows:

Theorem 1 *Let x^* be an optimal solution of problem*

$$\min_{x\in\mathcal{X}} \left\{ c^\top x + \sum_{P\in\mathcal{P}^*} \mathcal{Q}\left(x, \mathbb{E}\left[\xi|P\right]\right) \cdot \mathbb{P}(P) \right\}, \tag{8}$$

where $\mathcal{P}^*$ is a partition of Ω such that for each $P \in \mathcal{P}^*$, the optimal dual variables of $\mathcal{Q}(x^*, \xi)$ for $\xi \in P$ satisfy conditions (7). Then, x^* is also an optimal solution of problem

$$\min_{x\in\mathcal{X}} \left\{ c^\top x + \mathbb{E}\left[\mathcal{Q}(x, \xi)\right] \right\}. \tag{9}$$

Proof We first show that the objective value of x^* in (8) has the same value as that in (9), so its value is an upper bound for the latter problem. The other inequality is obtained by the convexity of $\mathcal{Q}(x, \cdot)$ and Jensen's inequality.

By the laws of total expectation, we know that for any numerable partition $\mathcal{P}$ of Ω,

$$\mathbb{E}\left[\mathcal{Q}(x, \xi)\right] = \sum_{P\in\mathcal{P}} \mathbb{E}\left[\mathcal{Q}\left(x, \xi\right)|P\right] \cdot \mathbb{P}(P).$$

In particular, for x^* and $\mathcal{P}^*$, according to Proposition 2, we obtain

$$c^\top x^* + \mathbb{E}\left[\mathcal{Q}(x^*, \xi)\right] = c^\top x^* + \sum_{P \in \mathcal{P}^*} \mathcal{Q}\left(x^*, \mathbb{E}\left[\xi \,|\, P\right]\right) \cdot \mathbb{P}(P).$$

Hence,

$$\min_{x \in \mathcal{X}} \left\{ c^\top x + \mathbb{E}\left[\mathcal{Q}(x, \xi)\right] \right\} \leq \min_{x \in \mathcal{X}} \left\{ c^\top x + \sum_{P \in \mathcal{P}^*} \mathcal{Q}\left(x, \mathbb{E}\left[\xi \,|\, P\right]\right) \cdot \mathbb{P}(P) \right\}.$$

Moreover, if $\hat{x}$ is an optimal solution of Problem (1), then

$$\begin{aligned} c^\top \hat{x} + \mathbb{E}\left[\mathcal{Q}(\hat{x}, \xi)\right] &= c^\top \hat{x} + \sum_{P \in \mathcal{P}^*} \mathbb{E}\left[\mathcal{Q}\left(\hat{x}, \xi\right) | P\right] \cdot \mathbb{P}(P) \\ &\geq c^\top \hat{x} + \sum_{P \in \mathcal{P}^*} \mathcal{Q}\left(\hat{x}, \mathbb{E}\left[\xi \,|\, P\right]\right) \cdot \mathbb{P}(P) \\ &\geq \min_{x \in \mathcal{X}} \left\{ c^\top x + \sum_{P \in \mathcal{P}^*} \mathcal{Q}\left(x, \mathbb{E}\left[\xi \,|\, P\right]\right) \cdot \mathbb{P}(P) \right\}, \end{aligned}$$

where the first equality is true by the laws of total expectation and the second inequality is given by Jensen's inequality and the convexity of $\mathcal{Q}(\hat{x}, \cdot)$. □

Note that this partition always exists, as presented in the following corollary.

Corollary 1 *If x^* is an optimal solution of problem*

$$\min_{x \in \mathcal{X}} \left\{ c^\top x + \mathbb{E}\left[\mathcal{Q}(x, \xi)\right] \right\}$$

then there exists a finite partition $\mathcal{P}^*$ of Ω such that

$$c^\top x^* + \mathbb{E}\left[\mathcal{Q}(x^*, \xi)\right] = c^\top x^* + \sum_{P \in \mathcal{P}^*} \mathcal{Q}\left(x^*, \mathbb{E}\left[\xi \,|\, P\right]\right) \cdot \mathbb{P}(P)$$

Proof The proof comes from the fact that the feasible set of the dual problem is the same for all ξ, so their optimal dual solutions must be in one of its extreme points, inducing an element of $\mathcal{P}^*$ for each vertex. In fact, note that the dual feasible solutions λ of $Q(x^*, \xi)$ must satisfy $W^\top \lambda \leq q$. Hence, for each $\xi \in \Omega$, we have an associated extreme point of $W^\top \lambda \leq q$ that is an optimal dual solution of $Q(x^*, \xi)$. This result induces a finite partition $\mathcal{P}^*$ of Ω such that all $\xi \in P$, $P \in \mathcal{P}^*$ have the same dual optimal solution of $Q(x^*, \xi)$. Hence, all values satisfy the conditions of Proposition 2, and

$$c^\top x^* + \mathbb{E}\left[\mathcal{Q}(x^*, \xi)\right] = c^\top x^* + \sum_{P \in \mathcal{P}^*} \mathbb{E}\left[\mathcal{Q}\left(x^*, \xi\right) | P\right] \cdot \mathbb{P}(P)$$

$$= c^\top x^* + \sum_{P \in \mathcal{P}^*} \mathcal{Q}\left(x^*, \mathbb{E}\left[\xi | P\right]\right) \cdot \mathbb{P}(P). \qquad \square$$

We note that some differences exist between our approach and the original APM proposed in [34]. The most relevant aspect of our approach is the possibility of extending it to the case of continuous distribution for the uncertainty space. Moreover, the condition proposed in the original paper to aggregate scenarios is such that all dual variables λ^ξ for $\xi \in P$ in each subset $P \subset \Omega$ must have the same value. This is a particular case that satisfies the conditions of Proposition 2 by means of the linearity of the expected value. Finally, [34] establishes that this criterion is required to have the equality between the value of the aggregated problem and the expected value of the atomized subproblems. Nonetheless, the presented conditions of Proposition 2 provide a framework where less demanding conditions might be applied to aggregate/disaggregate scenarios, e.g., degenerated subproblems with multiple optimal dual solutions.

4 Algorithm implementation

The method is presented in Algorithm 1. The idea of the method is to iteratively construct a partition $\mathcal{P}$ satisfying the conditions of Proposition 2. Initially, we start with a trivial partition ($\mathcal{P} = \{\Omega\}$) and split the partition based on the duals of the subproblems. This split procedure is problem dependent. At each iteration, the algorithm provides a lower bound (the optimal value of the aggregated problem) and, potentially, an upper bound. The upper bound can be computed by solving the expected value of the subproblem, which is easy to compute in the discrete case by solving the subproblem for each scenario independently. However, the computation can be difficult for continuous distributions. Nevertheless, if an upper bound cannot be provided, the algorithm can still be applied until the disaggregated partition satisfies the conditions of Proposition 2.

The key step in the generalized adaptive partition-based method implementation proposed in Algorithm 1 is how to execute Line 11 correctly. In the case of finite distribution $\Omega = \{\xi^s\}$, we can simply solve subproblems (2) for each scenario ξ^s and disaggregate P into its subsets satisfying conditions (7). For general continuous distributions, this step is fine-tuned according to the structure of subproblems (2). We discuss this case in Sect. 5, showing a few examples of how to execute this disaggregation.

We note that even if Proposition 2 requires that subproblem $\mathcal{Q}(x, \xi)$ be feasible for all $\xi \in P$, we do not require subproblem feasibility for all $x \in \mathcal{X}$. If $\mathcal{Q}(x, \xi)$ is infeasible for a subset $P' \subseteq P$, then we can disaggregate the partition splitting P into two subsets P' and $P \setminus P'$ and iterate in order to obtain a new first-stage solution.

Another approach to contend with infeasibilities and simultaneously try to satisfy conditions (7) is to split P using a dual extreme ray of the cone $\{\lambda^\xi \mid W^\top \lambda^\xi \leq 0\}$. Assuming that the dual subproblem $\mathcal{Q}^D(\bar{x}, \xi)$ is feasible (and then unbounded for an infeasible primal subproblem $\mathcal{Q}(\bar{x}, \xi)$), we can obtain an extreme ray by solving the problem

$$\mathcal{Q}^f(\bar{x}, \xi) := \max \left\{ (h^\xi - T^\xi \bar{x})^\top \theta^\xi \mid W^\top \theta^\xi \leq 0,\ e^\top \theta^\xi \leq 1 \right\}$$

Algorithm 1 An iterative implementation of the generalized APM

Require: A stopping threshold ϵ and an initial partition $\mathcal{P}^{(0)}$ of Ω
Ensure: An optimal partition $\mathcal{P}^*$ to attain an optimal solution $x^\star$ of Problem (1)
1: Set $t := 0$, $z_L^{(0)} := -\infty$ and $z_U^{(0)} := \infty$.
2: **loop**
3: $t = t + 1$
4: Solve Problem (4) for partition $\mathcal{P}^{(t)}$ and assign its optimal value to the lower bound $z_L^{(t)}$ and its optimal solution to $\bar{x}^{(t)}$
5: If possible, compute the upper bound $z_U^{(t)} := c^\top \bar{x}^{(t)} + \mathbb{E}[\mathcal{Q}(\bar{x}^{(t)}, \xi)]$
6: **if** $z_U^{(t)} - z_L^{(t)} < \epsilon$ **then**
7: **exit**
8: **end if**
9: **for all** $P \in \mathcal{P}^{(t)}$ **do**
10: **if** P does not satisfy conditions (7) for $\bar{x}^{(t)}$ **then**
11: Disaggregate P to satisfy conditions (7) and construct $\mathcal{P}^{(t+1)}$.
12: **end if**
13: **end for**
14: **if** $\mathcal{P}^{(t+1)} = \mathcal{P}^{(t)}$ **then**
15: **exit**
16: **end if**
17: **end loop**
18: **return** optimal solution $x^\star := \bar{x}^{(t)}$, optimal partition $\mathcal{P}^\star := \mathcal{P}^{(t)}$ and optimal value $z_L^{(t)}$

Similar to Corollary 1, since the feasible region of $\mathcal{Q}^f(\bar{x}, \xi)$ has a finite number of extreme points, we can disaggregate P into subsets that share the same optimal solution θ^ξ. Note that this approach always produces a splitting of P. In fact, if the same solution θ^* of $\mathcal{Q}^f(\bar{x}, \xi)$ is optimal for all $\xi \in P$, then θ^* is also optimal for $\mathcal{Q}^f(\bar{x}, \mathbb{E}[\xi|P])$, so $\mathcal{Q}(\bar{x}, \mathbb{E}[\xi|P])$ is infeasible, which is not possible because $\bar{x}$ comes from Problem (4) considering P as an element of the partition $\mathcal{P}^{(t)}$. We also note that the assumption of feasibility of the dual problem $\mathcal{Q}^D(\bar{x}, \xi)$ is implied, for example, by the complete recourse conditions for TSSPs.

To show an example, let us apply the algorithm to the problem $\min x + \mathbb{E}[\mathcal{Q}(x, \xi)]$ with $\mathcal{Q}(x, \xi) = \min\{y | y \leq x \wedge x \geq \xi\}$ and $\xi \rightsquigarrow U[0, 1]$. It is easy to see that $\mathcal{Q}(x, \xi)$ is infeasible for $\xi > x$ when $x < 1$, but the dual problem $\mathcal{Q}^D(x, \xi) = \max\{x + (\xi - x)\lambda | \lambda \geq 1\}$ has solution $\lambda = 1$ for $\xi \leq x$ and $\lambda = \infty$ in the other case. Hence, we can disaggregate the partition in each step intersecting its elements with the segments $[0, \bar{x}^{(t)}]$ and $[\bar{x}^{(t)}, 1]$. In this case, the first-stage solution in step t is given by $\bar{x}^{(t)} = 1 - 1/2^{t+1}$, which converges to the optimal solution $x^* = 1$.

We also highlight that the optimal solution $\bar{x}^{(t)}$ of Problem (4) for a partition $\mathcal{P}^{(t)}$ is also a feasible solution for partition $\mathcal{P}^{(t+1)}$, so the lower bound $z_L^{(t)}$ does not decrease in consecutive iterations. In the case of a finite distribution, the convergence is guaranteed because in each iteration, we increase the size of $\mathcal{P}^{(t)}$, which is upper bounded by $|\Omega|$. As our previous example exposes, we cannot ensure a finite convergence for continuous distributions. However, our computational experiments show that the algorithm quickly converges to a near-optimal solution.

5 Numerical experiments

Since the fundamental novelty of our proposal arises when stochastic parameters have continuous probability distributions, the computational experiments are designed to enlighten algorithmic behavior on two problems with this type of uncertainty from classic literature on stochastic programming. For the case of a discrete distribution, we refer the reader to the papers presented in the literature review.

We have divided the computational experiment into three parts. First, we discuss the implementation and results for a classic problem from the stochastic programming literature, the LandS instance, wherein uncertainty is presented in the right-hand side coefficients. The second problem is the TSSP reformulation of conditional value-at-risk (CVaR) minimization, where the uncertainty appears in the technological coefficient of the first-stage variables x. Both problems have well-defined structures that are useful to define the procedure to split the uncertainty space Ω at each iteration of the algorithm. Finally, the third problem is the fixed-charge multicommodity flow problem under stochastic demand with a discrete distribution, to evaluate the impact of the generalized conditions of Proposition 2 in comparison to the conditions of [34].

5.1 Energy planning problem—LandS

LandS, a classic problem in stochastic programming that is studied for academic purposes, was originally proposed in [19]. LandS is an energy planning investment problem, where the goal is to decide the capacities of four new plants while minimizing allocation and operational costs. The set of power plants is supposed to meet uncertain demand of three different electric modes. In the first stage, some minimum capacities and budget constraints must be satisfied; during the second stage, energy is distributed according to the realization of the uncertain demands. The mathematical formulation is as follows:

$$\min_{x \geq 0} \left\{ \sum_{i \in \mathbb{I}} c_i x_i + \mathbb{E}\left[\mathcal{Q}(x, \xi)\right] : \sum_{i \in \mathbb{I}} x_i \geq m, \sum_{i \in \mathbb{I}} c_i x_i \leq b \right\}$$

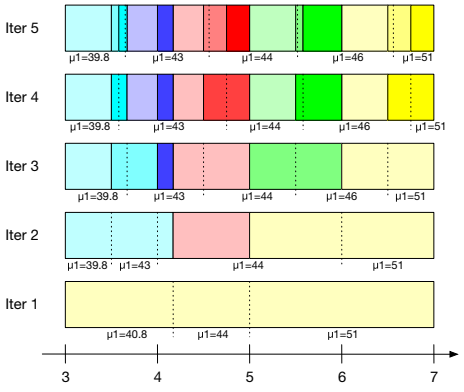
where

$$\mathcal{Q}(x, \xi) := \min_{y \geq 0} \sum_{i \in \mathbb{I}} \sum_{j \in \mathbb{J}} f_{ij} y_{ij} \tag{10a}$$

$$\text{s.t.} \sum_{j \in \mathbb{J}} y_{ij} \leq x_i, \quad \forall\, i \in \mathbb{I} \tag{10b}$$

$$\sum_{i \in \mathbb{I}} y_{ij} \geq d_j^{\xi}, \quad \forall\, j \in \mathbb{J} \tag{10c}$$

The original problem sets up an uncertain demand for d_1^{ξ} with three scenarios: 3, 5 or 7 units. In this experiment, we assume that d_1^{ξ} follows a uniform distribution in



Iter	x_1	x_2	x_3	x_4
1	0.833	3.000	4.167	4.000
2	2.500	3.000	3.500	3.000
3	1.833	4.000	3.667	2.500
4	2.000	4.167	3.583	2.250
5	1.917	4.083	3.625	2.375
6	1.875	4.042	3.646	2.438

Iter	LB	UB	Gap
1	378.667	382.711	1.0567%
2	380.122	381.100	0.2567%
3	380.601	380.844	0.0640%
4	380.842	380.893	0.0007%
5	380.843	380.856	0.0004%
6	380.844	380.847	0.0002%

Fig. 1 Iteration details for the LandS example: partition of Ω (left), current solution and objective bounds (right) at each iteration

the interval [3,7], following the ideas from [18]. Remaining demands are considered to be deterministic.

To split the uncertainty space Ω and compute an upper bound for the optimal value of the problem, we introduce the dual of $\mathcal{Q}(\hat{x}, \xi)$ given by

$$\mathcal{Q}^D(\hat{x}, \xi) := \max_{\nu, \mu \geq 0} \sum_{j \in \mathbb{J}} \mu_j d_j^{\xi} - \sum_{i \in \mathbb{I}} \nu_i \hat{x}_i$$
$$\mu_j - \nu_i \leq f_{ij} \qquad \forall i \in \mathbb{I}, j \in \mathbb{J}$$

where ν and μ correspond to the dual variables of constraints (10b) and (10c), respectively. Given an optimal solution of the subproblem for a given value of d_1^{ξ}, we can use sensitivity analysis to compute a neighborhood around d_1^{ξ} in which the dual optimal variables do not change. Moreover, $\mathcal{Q}^D(x, \xi)$ is a nondecreasing piecewise linear function on d_1^{ξ}, so the upper bound of Line 5 is easy to compute.

In our experiment, we start with $\mathcal{P}^{(0)} = \{[3, 7]\}$, and in each iteration, the partition is refined by dividing the corresponding elements of $\mathcal{P}$, utilizing the segment extremes of piecewise linear function $\mathcal{Q}^D(\bar{x}^{(t)}, \xi)$.

In Fig. 1, we show the resulting first six iterations of the algorithm. Columns LB and UB present the current lower bound (objective value of the aggregated problem) and the upper bound (computed by the benefit of $\bar{x}^{(t)}$ and subproblem optimal dual variables), respectively. The column Gap shows the relative gap between the current solution and the best upper bound obtained so far. After a few iterations, we obtain near-optimal solutions for the problem, with a gap close to the computational precision of the optimization software. In Fig. 1 (left), we present the partition in each iteration (highlighted by different colors) as well as the segments (dotted lines) obtained after carrying out the sensitivity analysis. The value under each segment corresponds to the dual variable of the stochastic demand constraint. Notably, these dual values do not change after iteration 3, but the extremes of the corresponding intervals change slightly in each iteration until converging to an optimal solution.

5.2 Conditional value-at-risk linear problems

A classic problem in risk optimization is to minimize the CVaR, which is a well-known risk measure satisfying the properties of coherency [2]. In our case, we assume a linear problem, where the objective coefficients $\tilde{r}^{\xi}$ are random, and we minimize $\text{CVaR}_{\delta}(x^{\top}\tilde{r}^{\xi})$ subject to linear constraints $Ax \leq b$, where δ indicates the risk aversion of the decisionmaker. This problem (see [28]) can be reformulated as

$$\min_{x,\tau}\left\{\tau + \frac{1}{\delta}\mathbb{E}\left[-x^{\top}\tilde{r}^{\xi} - \tau\right]^{+} : Ax \leq b\right\}$$

In our context, x and τ are the first-stage decisions, while the second-stage subproblem is

$$\mathcal{Q}((x,\tau),\xi)) := (-x^{\top}\tilde{r}^{\xi} - \tau)^{+} = \min\{z : z \geq -x^{\top}\tilde{r}^{\xi} - \tau, z \geq 0\}$$

Let us note that the dual of $\mathcal{Q}((x,\tau),\xi))$ has a single dual variable λ, and it can be formulated as

$$\max_{\lambda}\left\{(-x^{\top}\tilde{r}^{\xi} - \tau)\cdot\lambda : \lambda \leq 1, \lambda \geq 0\right\}.$$

Hence, an optimal solution of this dual problem is

$$\lambda^{*} = \begin{cases} 1 & \text{if } -\bar{x}^{\top}\tilde{r}^{\xi} - \bar{\tau} \geq 0 \\ 0 & \text{if not} \end{cases}$$

In other words, there is a hyperplane separating Ω, where the dual variables of the subproblem $\mathcal{Q}((\bar{x},\bar{\tau}),\xi)$ have the same value for a given pair $(\bar{x},\bar{\tau})$.

Therefore, from a partition $\mathcal{P}^{(t)}$ of Ω, we can compute $r^{P} = \mathbb{E}[\tilde{r}^{\xi}|P]$ and $p^{P} = \mathbb{P}(P)$ for all $P \in \mathcal{P}^{(t)}$ and solve the aggregated problem

$$\min_{x,\tau}\left\{\tau + \frac{1}{\delta}\sum_{P\in\mathcal{P}^{(t)}} p^{P}\cdot z_{P} : Ax \leq b, z_{P} \geq -x^{\top}r^{P} - \tau, z_{P} \geq 0\ \forall P \in \mathcal{P}^{(t)}\right\}$$

Given an optimal solution $(\bar{x}^{(t)},\bar{\tau}^{(t)})$ of this problem, we can split each $P \in \mathcal{P}^{(t)}$ into subsets $P' = P\bigcap\{\xi : -\bar{x}^{(t)\top}\tilde{r}^{\xi} \geq \bar{\tau}^{(t)}\}$ and $P'' = P\bigcap\{\xi : -\bar{x}^{(t)\top}\tilde{r}^{\xi} \leq \bar{\tau}^{(t)}\}$ to obtain a new partition.

Case study

For the computational test, we solve the classic portfolio problem, where x represents the fraction of the portfolio assigned to each investment, and the constraints of the first stage are $x^{\top}e = 1, x \geq 0$, ensuring to invest the whole budget in nonnegative fractions. Additionally, we assume that returns $\tilde{r}$ of each investment follow a multivariate normal

Table 1 Results for the CVaR portfolio example

Iter	LB	UB	Gap (%)	$\lvert\mathcal{P}^{(t)}\rvert$	x_1	x_2
1	−0.0702	0.7641	109.184	1	0	1
2	0.0408	0.6054	93.2602	2	1	0
3	0.3196	0.6054	47.2124	4	1	0
4	0.3585	0.7641	40.7887	6	0	1
5	0.4584	0.5104	10.1866	9	0.59	0.41
6	0.5001	0.5222	2.0277	14	0.7752	0.2248
7	0.5043	0.5095	1.0259	20	0.6834	0.3166
8	0.5070	0.5082	0.2305	27	0.6371	0.3629
9	0.5082	0.5082	0.0039	34	0.6375	0.3625

distribution $\tilde{r}^{\xi} \rightsquigarrow \mathcal{N}(\mu, \Sigma)$ using historical data for stocks listed on the S&P 500, as in [16,37].

Note that in each iteration, given $(\bar{x}^{(t)}, \bar{\tau}^{(t)})$, we can compute an upper bound for the problem expressed as

$$\text{CVaR}_{\delta}(\bar{x}^{(t)\top}\tilde{r}^{\xi}) := \mu^{\top}\bar{x}^{(t)} + \tfrac{\sigma}{\delta}\phi(\Phi^{-1}(\delta)) \tag{11}$$

where $\sigma = \bar{x}^{(t)\top}\Sigma\bar{x}^{(t)}$ and ϕ and Φ are the standard normal probability density function and standard normal quantile, respectively.

We solve the problem using two stocks and a risk level of $\delta = 0.1$ to provide a graphical representation of the algorithm. To estimate the probabilities and expected return of each region, we use a Monte Carlo sampler of the underlying distribution. Table 1 shows the results for our instance. We can notice that the problem converges quickly to an optimal solution, as well as in the previous LandS example. A more detailed analysis can be seen in Fig. 2, where the region Ω is presented, with ellipses indicating the 50%, 80%, 90%, 95% and 99% confidence intervals of the normal bidimensional distribution. In the first four iterations, the aggregated model considers only the riskiest scenario (bottom-left dot) and invests the entire portfolio in the stock with the highest return r'. Our algorithm generates a cut that divides the uncertainty region into $r_i^{\xi} \geq r'$ and $r_i^{\xi} \leq r'$, where i is the stock where the budget is invested. After the fifth iteration, the portfolio starts to combine stocks, and the region of interest is divided more precisely to obtain a better estimation of the optimal problem solution.

Comparison with the sample average approximation technique

We use this example to compare our proposed method to the classical sample average approximation (SAA) technique for TSSP [15]. This technique is very popular because it allows to approximate the stochastic programming problem for a continuous probability distribution by a problem with a discrete distribution of samples from the original distribution. Moreover, it is known that for a sufficiently large number of samples, both the optimal value and the optimal solution of the SAA problem converge to the optimal value and solution of the continuous distribution [15,32]. The main problem with SAA

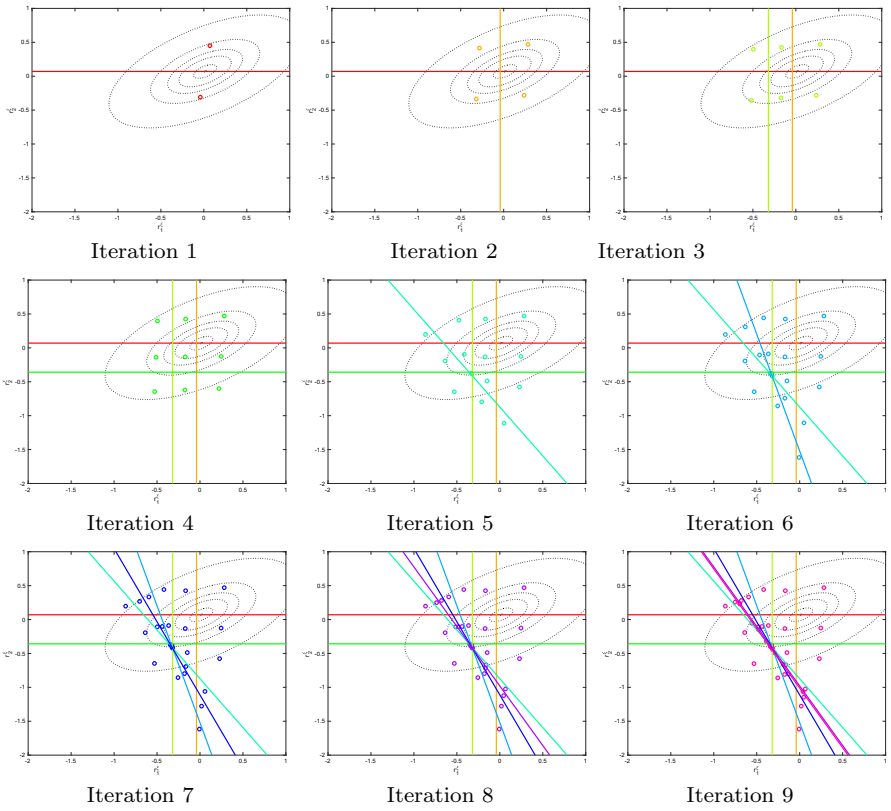
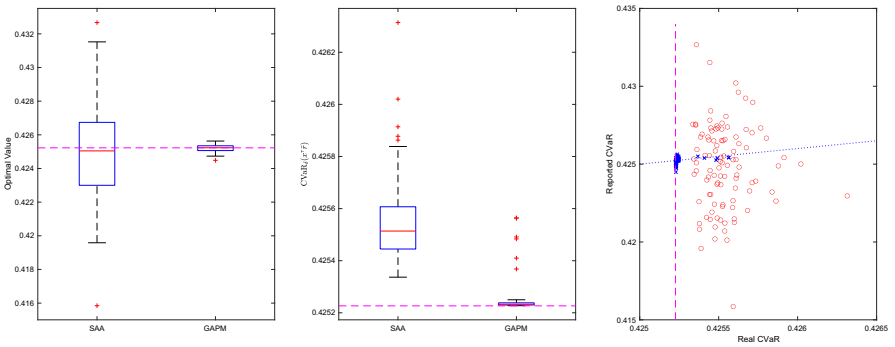


Fig. 2 Partition of Ω at each iteration for the CVaR portfolio example

is that it strongly relies on an appropriate representation of the original uncertainty space by these sampled scenarios. Furthermore, if too many scenarios are generated, then the resulting optimization problem is intractable; hence, different techniques have been studied to improve its performance [13].

To compare both approaches, we solve the same problem with 10 stocks and a risk level of $\delta = 0.01$. For the SAA case, we sample 100,000 scenarios from the distribution, and we solve the SAA problem up to optimality. For GAPM, we apply the algorithm until either the gap is smaller than 1E–4 or until the partition does not further refine. In both cases, we record the *Reported CVaR* as the final objective value reported and the *Real CVaR* as the real value of the optimal solution of each problem, calculated using equation (11). We repeat this procedure 100 times with a new set of samples for both methods to evaluate the sensitivity of each method to different samples.

Figure 3 presents boxplots of the obtained values for each method. We also include a dashed magenta line indicating the true optimal CVaR value for the problem. It can be seen that the solution of the SAA method changes considerably depending on the initial sample provided to solve the problem. This occurs because the method should approximate the uncertainty space of the CVaR problem (in this case, an ellipsis) by



(a) Reported CVaR **(b)** Real CVaR **(c)** Real vs Reported CVaR

Fig. 3 Resulting objective values for SAA and GAPM methods

a convex hull of the scenarios, which is particularly difficult for higher dimensions and lower risk levels (see [16] for a discussion on this behavior). This also impacts the true CVaR value of the optimal solutions found, which have a smaller dispersion but no correlation between the reported and true CVaR values. This correlation can be observed in Fig. 3c, where the reported and true CVaR value of each run is presented as a red dot (SAA) or a blue cross (GAPM), and a blue line indicates the identity. This is a relevant issue because it indicates that a single run of the SAA method is unreliable, especially as the dimension of the problem increases, and approximation of the uncertainty space by a convex hull of samples becomes increasingly difficult. In contrast, GAPM provides a more robust approach for the problem with less dependency on the samples. Furthermore, the true CVaR values of the resulting solutions have lower dispersion, and their values are closer to the true optimal value of the problem. In fact, GAPM depends only on samples to evaluate the conditional expected values and probabilities of the different regions defined by the hyperplanes, which seems to produce more stable results than those for the optimization problem. Over the 100 runs, GAPM requires 15 to 24 iterations to finish, and the final partition size $|\mathcal{P}|$ has between 4296 and 9044 scenario subsets, which is considerably smaller than the 100,000 scenarios utilized by SAA to solve the problem.

It is important to note that the computational burden of applying GAPM does not come from the optimization problem itself but rather from the simulation techniques required to estimate the conditional expected values of each region in $\mathcal{P}$. In these computational experiments, we implemented a naive Monte Carlo simulation with a fixed set of simulations for all iterations. This step consumes most of the time required by the algorithm. This is relevant for larger dimensions of the uncertainty space because, similar to the SAA method, GAPM can suffer from the *curse of dimensionality*, resulting in more iterations and smaller regions of the uncertainty space and thus more difficulty to sample correctly. A proper implementation of the algorithm requires specialized simulation techniques to estimate these values properly, which are known only for some particular distributions (e.g., see [7] for the Gaussian case), which goes beyond the scope of this paper. Nevertheless, our experiment shows that GAPM works sufficiently well, even with a simple naive Monte Carlo estimation.

5.3 Stochastic fixed-charge multicommodity flow problem

The previous examples aggregate scenarios with the same vector of optimal dual values for the subproblem. In this example, we show the impact of applying the conditions of Proposition 2 to decide how to split the partition in each step of GAPM.

The stochastic fixed-charge multicommodity flow problem (SFCMCF) is a well-known problem in network design, where different methods have been developed to solve the stochastic case, most of which were based on Benders cuts [26]. In this problem, a first-stage decision is taken to construct arcs in a network with a predefined capacity, and a multicommodity flow must be routed through the constructed network minimizing the construction costs plus expected cost of the routing. Let $G = (V, E)$ be the potential network to be constructed, where each arc $(i, j) \in E$ has a construction cost f_{ij} and a capacity u_{ij}. Let $\mathcal{K}$ be a set of commodities, where each commodity $k \in \mathcal{K}$ has a random demand $\tilde{d}_k$ that must be routed from an origin node $O(k)$ to a destination node $D(k)$ with a unitary cost of c_{ijk} in each arc $(i, j) \in E$. The problem is then formulated as

$$\min_{x \in \mathcal{X}} \sum_{(i,j) \in E} f_{ij} x_{ij} + \mathbb{E}[\mathcal{Q}(x, \xi)]$$

where

$$\mathcal{Q}(x, \xi) := \min_{y \geq 0} \sum_{k \in \mathcal{K}} \sum_{(i,j) \in E} c_{ijk} y_{ijk} \tag{12a}$$

$$\text{s.t.} \sum_{j:(i,j) \in E} y_{ijk} - \sum_{j:(j,i) \in E} y_{ijk} = \begin{cases} d_k^\xi & i = O(k) \\ -d_k^\xi & i = D(k) \\ 0 & \text{other case} \end{cases} \quad \forall i \in V, k \in \mathcal{K} \tag{12b}$$

$$\sum_{k \in \mathcal{K}} y_{ijk} \leq u_{ij} x_{ij} \quad \forall (i, j) \in E \tag{12c}$$

First-stage decision variables x are binary variables indicating if arc $(i, j) \in E$ is constructed. Second-stage decision variables y_{ijk} are the flow of commodity k routed through arc $(i, j) \in E$. Constraints (12b) are the classic flow constraints for each commodity, and constraints (12c) are capacity constraints in each constructed arc over the total flow of all commodities.

In this case, we can provide a simpler condition for aggregating scenarios due to Proposition 2. In fact, note that first-stage variables x do not have random coefficients on the subproblem, so conditions (7b) are always satisfied. Moreover, the right-hand side coefficients of the subproblem are equal to zero except for the origin and destination nodes of each commodity. Hence, conditions (7a) can be written as

$$\sum_{k\in\mathcal{K}} \mathbb{E}[d_k^\xi|P]\left(\mathbb{E}[\lambda^\xi_{O(k),k}|P] - \mathbb{E}[\lambda^\xi_{D(k),k}|P]\right) = \sum_{k\in\mathcal{K}} \mathbb{E}\left[d_k^\xi \cdot (\lambda^\xi_{O(k),k} - \lambda^\xi_{D(k),k})|P\right] \tag{13}$$

where λ^ξ_{ik} corresponds to the dual variable of constraints (12b) for $i \in V, k \in \mathcal{K}$.

The previous condition provides useful information on how to construct a partition $\mathcal{P}$ to apply Theorem 1. For example, the condition is satisfied if each $P \in \mathcal{P}$ has the same demands $(d_k^\xi)_{k\in\mathcal{K}}$ or the same vector of differences of dual variables $(\lambda^\xi_{O(k),k} - \lambda^\xi_{D(k),k})_{k\in\mathcal{K}}$ for each $\xi \in P$. Note that these conditions require comparing a vector of size $|\mathcal{K}|$ among the scenarios, which is considerably smaller than comparing the complete vector of dual variables, which has size $|V| \cdot |\mathcal{K}| + |E|$.

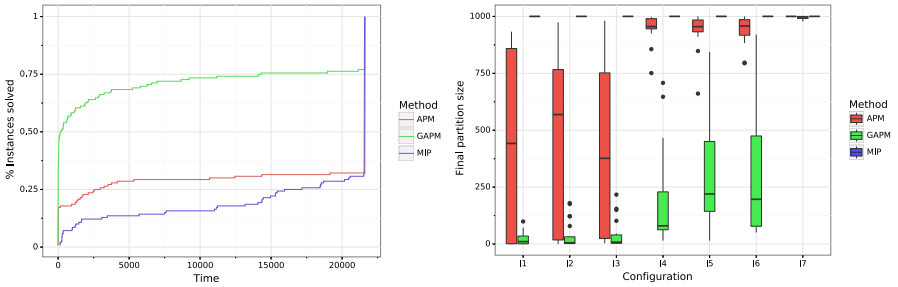
If the subproblem is infeasible (which occurs when the arcs' capacities are not sufficient to route all the demand in the scenario), we disaggregate the partition using extreme rays of the subproblems, as mentioned in Sect. 4.

Case study

We test our conditions on the Canad instances **R** from ORlib [4]. In particular, we consider the stochastic version of these instances presented in [26]. In this experiment, we use instances with a discrete distribution of 1000 scenarios. We apply our algorithm over 4 instance classes (r04, r05 and r07, r08), with five different levels of correlations among commodities and seven different cost/capacity configurations (I1 to I7). We omit instances r06 and r09 (with 50 commodities) and configurations I8 and I9 because none of the implemented methods are able to solve these problems efficiently.

We compare three different approaches: First, MIP corresponds to the extended formulation of the problem. Second, GAPM corresponds to Algorithm 1 disaggregating in each iteration among the scenarios with the same differences of dual variables $(\lambda^\xi_{O(k),k} - \lambda^\xi_{D(k),k})_{k\in\mathcal{K}}$. Finally, APM also corresponds to Algorithm 1 but disaggregates the partition among the scenarios with the same complete vector of dual variables, as originally presented in [34].

Figure 4a shows the percentage of instances solved up to optimality over time, with a time limit of 6 h. It can be seen that the adaptive partition method outperforms the extended formulation of the problem, particularly solving a larger percentage of instances in shorter times. Moreover, it can be seen that our generalized conditions presented in Proposition 2 and (13) allow to considerably improve the performance of the algorithm, when compared with the former conditions presented in [34]. Figure 4b provides insights to understand this behavior. The figure shows boxplots with the size of $|\mathcal{P}|$ at the end of the algorithm. It can be seen that APM requires a partition smaller than the number of scenarios for configurations $I1$, $I2$ and $I3$, allowing a faster solution of the problem than the extended MIP formulation. In contrast, for the other configurations, the required size for $\mathcal{P}$ is similar to the number of the scenarios, obtaining no benefit from running APM instead of MIP. This explains why the number of instances solved by MIP becomes closer to APM over time. The figure also shows



(a) Instances solved over time **(b)** Final partition size $|\mathcal{P}|$

Fig. 4 Performance of the different methods for stochastic fixed-charge multicommodity flow problems

that GAPM requires considerably smaller partition sizes to solve the problem, due to the tightened condition provided by Proposition 2. In fact, it can solve most of the instances with configurations $I1$ to $I6$. However, we can see that for configuration $I7$, in most of the cases, all scenarios are equally important to obtain the optimal solution, so no efficient aggregation can be performed.

Notes

Finally, we note that in both computational examples, several algorithmic improvements can be implemented to solve larger and more complex problems (e.g., reaggregating regions with the same duals, considering only the last k cuts, or subdividing only the active regions; see [21] for more details). Nevertheless, our purpose is simply to show how the method can automatically divide the uncertainty space to iteratively define the regions of interest for the problem and obtain an optimal solution. We also note that for problems with high dimensional uncertainty and different continuous distributions, it could be challenging to compute conditional expectations and element probabilities.

6 Conclusions

We present a generalization of the adaptive partition-based method for solving two-stage stochastic programming problems that contributes to extending the method to a more general setting, particularly, to consider continuous distributions of the uncertain parameters. The resulting algorithm allows for this type of problem to be addressed by automatically disaggregating the uncertainty space and solving a discrete (tractable) problem in each iteration. Computational experiments show the efficacy of the method to refine the uncertainty space in different regions of interest. It is important to remark that the refining procedure depends considerably on the structure of the problem, but it is sufficiently general for a broad family of problems. We are confident that this research represents a starting point for further development of computational methods for stochastic programming problems with continuous distributions.

References

1. Angulo, G., Ahmed, S., Dey, S.S.: Improving the integer L-shaped method. INFORMS J. Comput. **28**(3), 483–499 (2016). https://doi.org/10.1287/ijoc.2016.0695
2. Artzner, P., Delbaen, F., Eber, J.M., Heath, D.: Coherent measures of risk. Math. Finance **9**(3), 203–228 (1999)
3. Barik, S., Biswal, M.P., Chakravarty, D.: Two-stage stochastic programming problems involving some continuous random variables. J. Uncertain Syst. **7**(4), 277–288 (2013)
4. Beasley, J.E.: OR-library: distributing test problems by electronic mail. J. Oper. Res. Soc. **41**, 1069–1072 (1990). https://doi.org/10.2307/2582903
5. Bienstock, D., Zuckerberg, M.: Solving LP relaxations of large-scale precedence constrained problems. In: International Conference on Integer Programming and Combinatorial Optimization, pp. 1–14. Springer (2010). https://doi.org/10.1007/978-3-642-13036-6_1
6. Birge, J.R., Wets, R.J.B.: Designing approximation schemes for stochastic optimization problems. In: Particular for Stochastic Programs with Recourse, pp. 54–102. Springer, Berlin (1986). https://doi.org/10.1007/BFb0121114
7. Botev, Z.I.: The normal law under linear restrictions: simulation and estimation via minimax tilting. J. R. Stat. Soc. Ser. B (Stat. Methodol.) **79**(1), 125–148 (2017). https://doi.org/10.1111/rssb.12162
8. Crainic, T.G., Maggioni, F., Perboli, G., Rei, W.: Reduced cost-based variable fixing in two-stage stochastic programming. Ann. Oper. Res. (2018). https://doi.org/10.1007/s10479-018-2942-8
9. Edirisinghe, N.C.P., Ziemba, W.T.: Tight bounds for stochastic convex programs. Oper. Res. **40**(4), 660–677 (1992). https://doi.org/10.1287/opre.40.4.660
10. Edmundson, H.: Bounds on the expectation of a convex function of a random variable. Tech. rep., The RAND Corporation (1956)
11. Espinoza, D., Moreno, E.: A primal-dual aggregation algorithm for minimizing conditional value-at-risk in linear programs. Comput. Optim. Appl. **59**(3), 617–638 (2014). https://doi.org/10.1007/s10589-014-9692-6
12. Higle, J.L., Sen, S.: Stochastic decomposition: an algorithm for two-stage linear programs with recourse. Math. Oper. Res. **16**(3), 650–669 (1991). https://doi.org/10.1287/moor.16.3.650
13. Homem-de-Mello, T., Bayraksan, G.: Monte Carlo sampling-based methods for stochastic optimization. Surv. Oper. Res. Manag. Sci. **19**(1), 56–85 (2014). https://doi.org/10.1016/j.sorms.2014.05.001
14. Kall, P., Mayer, J.: Stochastic Linear Programming. International Series in Operations Research & Management Science, vol. 156, 2nd edn. Springer, Berlin (2011)
15. Kleywegt, A.J., Shapiro, A., Homem-de-Mello, T.: The sample average approximation method for stochastic discrete optimization. SIAM J. Optim. **12**(2), 479–502 (2002). https://doi.org/10.1137/S1052623499363220
16. Lagos, G., Espinoza, D., Moreno, E., Vielma, J.P.: Restricted risk measures and robust optimization. Eur. J. Oper. Res. **241**(3), 771–782 (2015). https://doi.org/10.1016/j.ejor.2014.09.024
17. Lemaréchal, C., Nemirovskii, A., Nesterov, Y.: New variants of bundle methods. Math. Program. **69**(1–3), 111–147 (1995). https://doi.org/10.1007/BF01585555
18. Linderoth, J., Shapiro, A., Wright, S.: The empirical behavior of sampling methods for stochastic programming. Ann. Oper. Res. **142**(1), 215–241 (2006). https://doi.org/10.1007/s10479-006-6169-8
19. Louveaux, F.V., Smeers, Y.: Optimal investments for electricity generation: a stochastic model and a test problem. In: Ermoliev, Y., Wets, R.J.B. (eds.) Numerical Techniques for Stochastic Optimization Problems. Springer, Berlin (1988)
20. Madansky, A.: Bounds on the expectation of a convex function of a multivariate random variable. Ann. Math. Stat. **30**(3), 743–746 (1959). https://doi.org/10.1214/aoms/1177706203
21. Muñoz, G., Espinoza, D., Goycoolea, M., Moreno, E., Queyranne, M., Letelier, O.R.: A study of the Bienstock–Zuckerberg algorithm: applications in mining and resource constrained project scheduling. Comput. Optim. Appl. **69**(2), 501–534 (2018). https://doi.org/10.1007/s10589-017-9946-1
22. Oliveira, W., Sagastizábal, C., Scheimberg, S.: Inexact bundle methods for two-stage stochastic programming. SIAM J. Optim. **21**(2), 517–544 (2011). https://doi.org/10.1137/100808289
23. Pay, B.S., Song, Y.: Partition-based decomposition algorithms for two-stage stochastic integer programs with continuous recourse. Ann. Oper. Res. (2017). https://doi.org/10.1007/s10479-017-2689-7
24. Pereira, M.V., Pinto, L.M.: Multi-stage stochastic optimization applied to energy planning. Math. Program. **52**(1–3), 359–375 (1991). https://doi.org/10.1007/BF01582895

25. Rahmaniani, R., Crainic, T., Gendreau, M., Rei, W.: A benders decomposition method for two-stage stochastic network design problems. Tech. Rep. CIRRELT-2017-22, Centre interuniversitaire de recherche sur les réseaux d'entreprise, la logistique et le transport (2017)
26. Rahmaniani, R., Crainic, T.G., Gendreau, M., Rei, W.: Accelerating the benders decomposition method: application to stochastic network design problems. SIAM J. Optim. **28**(1), 875–903 (2018). https://doi.org/10.1137/17M1128204
27. Rahmaniani, R., Crainic, T.G., Gendreau, M., Rei, W.: An asynchronous parallel benders decomposition method. Tech. Rep. CIRRELT-2019-49, Centre interuniversitaire de recherche sur les réseaux d'entreprise, la logistique et le transport (2019)
28. Rockafellar, R.T., Uryasev, S., et al.: Optimization of conditional value-at-risk. J. Risk **2**, 21–42 (2000). https://doi.org/10.21314/JOR.2000.038
29. Rockafellar, R.T., Wets, R.J.B.: Scenarios and policy aggregation in optimization under uncertainty. Math. Oper. Res. **16**(1), 119–147 (1991). https://doi.org/10.1287/moor.16.1.119
30. Ruszczyński, A.: A regularized decomposition method for minimizing a sum of polyhedral functions. Math. Program. **35**(3), 309–333 (1986). https://doi.org/10.1007/BF01580883
31. Ryan, K., Rajan, D., Ahmed, S.: Scenario decomposition for 0-1 stochastic programs: improvements and asynchronous implementation. In: 2016 IEEE International Parallel and Distributed Processing Symposium Workshops, pp. 722–729. IEEE (2016). https://doi.org/10.1109/IPDPSW.2016.119
32. Shapiro, A., Dentcheva, D., Ruszczyński, A.: Lectures on Stochastic Programming: Modeling and Theory. MOS-SIAM Series on Optimization. SIAM, Philadelphia (2014)
33. Siddig, M., Song, Y.: Adaptive partition-based SDDP algorithms for multistage stochastic linear programming. Tech. rep. (2019). arXiv:1908.11346
34. Song, Y., Luedtke, J.: An adaptive partition-based approach for solving two-stage stochastic programs with fixed recourse. SIAM J. Optim. **25**(3), 1344–1367 (2015). https://doi.org/10.1137/140967337
35. van Ackooij, W., de Oliveira, W., Song, Y.: Adaptive partition-based level decomposition methods for solving two-stage stochastic programs with fixed recourse. INFORMS J. Comput. **30**(1), 57–70 (2017). https://doi.org/10.1287/ijoc.2017.0765
36. Van Slyke, R.M., Wets, R.: L-shaped linear programs with applications to optimal control and stochastic programming. SIAM J. Appl. Math. **17**(4), 638–663 (1969). https://doi.org/10.1137/0117061
37. Vielma, J.P., Ahmed, S., Nemhauser, G.L.: A lifted linear programming branch-and-bound algorithm for mixed-integer conic quadratic programs. INFORMS J. Comput. **20**(3), 438–450 (2008). https://doi.org/10.1287/ijoc.1070.0256
38. Watson, J.P., Woodruff, D.L.: Progressive hedging innovations for a class of stochastic mixed-integer resource allocation problems. CMS **8**(4), 355–370 (2011). https://doi.org/10.1007/s10287-010-0125-4
39. Wolf, C., Fábián, C.I., Koberstein, A., Suhl, L.: Applying oracles of on-demand accuracy in two-stage stochastic programming—a computational study. Eur. J. Oper. Res. **239**(2), 437–448 (2014). https://doi.org/10.1016/j.ejor.2014.05.010
40. Zverovich, V., Fábián, C.I., Ellison, E.F., Mitra, G.: A computational study of a solver system for processing two-stage stochastic LPs with enhanced benders decomposition. Math. Program. Comput. **4**(3), 211–238 (2012). https://doi.org/10.1007/s12532-012-0038-z